brechungs-Aufruf im Sekundär-Betriebssystem (SBS) auslösen soll), und dass die Unterbrechungs-Aufruf-Tabelle zum Aufruf der entsprechenden Interrupt-Behandlungsroutine im Sekundär-Betriebssystem (SBS) ausgebildet ist.

24. Vorrichtung nach einem der Ansprüche 18 bis 23, **dadurch gekennzeichnet, dass** sie dazu ausgebildet ist, bei Aktivität des Sekundär-Betriebssystems (SBS) auf eine Unterbrechungsanforderung durch die in der Unterbrechungs-Aufruf-Tabelle des Sekundär-Betriebssystems gespeicherte Information, an welcher Stelle im Sekundär-Betriebssystem die Abarbeitung des Unterbrechungs-Aufrufs (Interrupts) zu erfolgen hat, die Unterbrechungs-Aufruf-Behandlungsroutine des Sekundär-Betriebssystems (SBS) direkt ohne Umweg über den System-Treiber, nur durch das Sekundär-Betriebssystem aufzurufen.

Claims

1. A method for operating (implementing) a secondary operating system on a processor in addition to a primary operating system, involving a switch from the primary operating system to the secondary operating system brought about by an interrupt call (interrupt), **characterised in that** a secondary operating system driver (SOS driver) of the primary operating system is loaded and activated for loading and controlling the secondary operating system, **in that** the interrupt tables of the operating systems are exchanged during the interrupt call and **in that** the secondary operating system driver determines by means of an interrupt service routine the information stored in the interrupt table of the secondary operating system as to the point in the secondary operating system at which the interrupt call is to be serviced.

2. A method according to claim 1, **characterised in that** memory contexts (virtual operating areas) are created in the computer's central processing unit (CPU)

3. A method according to either of claim 1 or claim 2, **characterised in that** a switch between operating systems proceeds by means of the SOS driver of the primary operating system and the board support package (BSP).

4. A method according to any one of claims 1 to 3, **characterised in that** the secondary operating system controls a switch to the primary operating system.

5. A method according to claim 4, **characterised in that** a switch from the secondary operating system to the primary operating system proceeds when the secondary operating system is idle (entry into idle loop).

6. A method according to claim 5, **characterised in that** a switch from the secondary operating system to the primary operating system proceeds by an instruction in the program sequence of the secondary operating system.

7. A method according to any one of the preceding claims, **characterised in that** the switch between operating systems proceeds by means of program code filed in a tunnel zone of the memory device.

8. A method according to any one of the preceding claims, **characterised in that** interrupt calls from the primary operating system are blocked while the secondary operating system is running.

9. A method according to any one of the preceding claims, **characterised in that** an interrupt service routine in the SOS driver reads the interrupt table of the secondary operating system and processing of the latter proceeds or is continued at the point to which the interrupt call relates.

10. A method according to any one of the preceding claims, **characterised in that** the system driver for each interrupt assigned to the secondary operating system (which is thus intended to trigger an interrupt call in the secondary operating system) generates an entry in the interrupt table in the primary operating system, which then in turn triggers a call to the corresponding interrupt service routine in the secondary operating system.

11. A method according to any one of the preceding claims, **characterised in that**, in the case of activity of the secondary operating system (SOS), in response to an interrupt request by the information stored in the interrupt table of the secondary operating system as to the point in the secondary operating system at which the interrupt call (interrupt) is to be serviced, the interrupt service routine of the secondary operating system (SOS) is called directly solely by the secondary operating system without passing via the system driver.

12. A method according to any one of claims 9 to 11, **characterised in that**, once a corresponding interrupt call has occurred and the point in the secondary operating system at which the interrupt is to be serviced has been determined, processing of the interrupt is continued in the secondary operating system at the point to which the interrupt call relates.

13. A method according to any one of the preceding

claims, **characterised in that**, on switching from one operating system to the other, all system states of the one operating system are stored.

14. A method according to any one of the preceding claims, **characterised in that**, on switching from one operating system to the other operating system, all system states of the other operating system are loaded.

15. A method according to any one of the preceding claims, **characterised in that** clock pulses for the secondary operating system are generated by the main hardware timer.

16. A method according to any one of the preceding claims, **characterised in that** clock pulses for the primary operating system are generated by a clock pulse system driver.

17. A device for operating a secondary operating system on a processor in addition to a primary operating system involving a switch from the primary operating system to the secondary operating system brought about by an interrupt call (interrupt), **characterised by** a secondary operating system driver (SOS driver) of the primary operating system for loading and controlling the secondary operating system, a function for exchanging the interrupt tables of the operating systems being provided for servicing the interrupt call and an interrupt service routine being provided in the SOS driver for determining the information stored in the interrupt table of the secondary operating system as to the point in the secondary operating system at which the interrupt call is to be serviced.

18. A device according to claim 17, **characterised in that** the SOS driver comprises a tunnel context updating routine to update a tunnel context in the central processing unit (CPU).

19. A device, in particular according to claim 17 or claim 18, **characterised in that** it is configured to exchange the interrupt tables in the event of a switch in the activity of the operating systems.

20. A device according to claim 17 or claim 19, **characterised in that** the SOS driver comprises an interrupt table change routine to generate entries in the interrupt table of the primary operating system, which at least makes entries for the interrupt calls for the secondary operating system.

21. A device according to any one of claims 17 to 20, **characterised in that** the board support package (BSP) comprises a section for return to the primary operating system (POS).

22. A device according to any one of claims 17 to 21, **characterised in that** the secondary operating system driver (SOS driver) comprises an interrupt tables section by means of which it generates an interrupt table in the primary operating system, which interrupt table contains a call for an interrupt service routine for calling the secondary operating system.

23. A device according to any one of claims 17 to 22, **characterised in that** the system driver is configured to generate an entry in the interrupt table in the primary operating system (POS) for each interrupt assigned to the secondary operating system (SOS) (which is thus intended to trigger an interrupt call in the secondary operating system (SOS)), and **in that** the interrupt table is configured to call the corresponding interrupt service routine in the secondary operating system (SOS).

24. A device according to any one of claims 18 to 23, **characterised in that** it is configured, in the case of activity of the secondary operating system (SOS), in response to an interrupt request by the information stored in the interrupt table of the secondary operating system as to the point in the secondary operating system at which the interrupt call (interrupt) is to be serviced, to call the interrupt service routine of the secondary operating system (SOS) directly solely by the secondary operating system without passing via the system driver.

Revendications

1. Procédé pour faire fonctionner (implémenter) un système d'exploitation secondaire sur un processeur à côté d'un système d'exploitation primaire, avec commutation du système d'exploitation primaire au système d'exploitation secondaire provoquée par un appel d'interruption (Interrupt), **caractérisé en ce que** pour charger et amorcer le système d'exploitation secondaire un pilote de système d'exploitation secondaire (pilote SES) du système d'exploitation primaire est chargé et activé, **caractérisé** également **en ce que** lors de l'appel d'interruption, un échange a lieu entre les tables d'interruption des systèmes d'exploitation, **caractérisé en ce que** le pilote du système d'exploitation secondaire détermine à l'aide d'une routine de traitement des appels d'interruption l'information enregistrée dans la table d'interruption du système d'exploitation secondaire indiquant à quel endroit du système d'exploitation secondaire doit être exécuté l'appel d'interruption.

2. Procédé selon la revendication 1, **caractérisé en ce que** dans l'unité centrale de l'ordinateur (CPU) sont créés des contacts de mémoire (espaces de travail virtuels).